VideOSC:
moving control from gesture to texture

Stefan Nussbaumer
Independent artist, Austria
stefan [at] basislager.org
http://pustota.basislager.org
Proceedings of Korean Electro-Acoustic Music Society's 2016 Annual Conference (KEAMSAC2016)
Seoul, Korea, 15-16 October 2016

VideOSC is an OSC[1] controller for mobile phones and tablet computers running on Googles Android operating system. Unlike many others it does not follow a conventional concept using sliders or knobs for manual interaction. It does not use any sensors like orientation or acceleration sensors. Its mere source for control is the incoming video stream of the devices inbuilt camera, using the RGB information of the pixels to control sound synthesis or other forms of electronic media.

The technical development during the last ten to fifteen years has not only created a new category of communication devices, tablet computers and smartphones. It also provided new technical platforms for media control in the field of artistic expression, be it visual, acoustic or otherwise. The invention of network based protocols like OSC has enabled a communication between devices over wireless networks, making performance independent from comprehensive hardware infrastructure.

Development Background

Smartphone- or tablet computer-based control applications nevertheless still seem to follow a very traditional scheme of manual interaction to establish control. Of course, physical knobs and sliders are gone. Yet they have often only been replaced by virtual ones, maybe amended by some sensors (orientation, accelerometers, etc.) which may loosen the rigid constraints of manual interaction in favour of more open gestural control over the instrument.

This is the point where VideOSC takes a different approach by using visual information as source of control. The application uses the video stream of a telephone's or tablet computer's inbuilt camera as source for OSC messages that are sent over a Wi-Fi network to a receiving computer or device that may use the data to control sound creation or other media. Of course that does not mean that the performer will become redundant in the process – still it is him or her who direct the camera, selects the image section and as a consequence defines control.

Significance of Visual Information

The crucial problem when working with visual information is the amount of information that comes in with every single frame from the video stream. Even a low resolution image, e. g. 320 x 240 pixels, will hold an amount of width x height x 3 distinct values of colour information which is on the one hand too much to be processed into OSC messages in a meaningful way, on the other hand no (digital) synthesis structure will probably need such an amount of data to control its parameters.

The German artist, programmer and theorist Julian Rohrhuber characterizes in his article "Operation, Operator – Sehen, was das Photon sieht" abstraction within scientific research as follows:

> "Symmetry and abstraction are two central as well as disputed elements within scientific representation. What they have in common is a peculiar, targeted indifference adverse to differences, an indifference which is by a lesser degree a sign of inexactness, but rather coins the credibility, the elegance or economy of scientific solutions". (Julian Rohrhuber, 2011: 73)

Though Rohrhuber is speaking about scientific methodology the pattern may be common to human perception in general. A blurry image of an object, as long as certain characteristics of the object are preserved, may be enough to clearly identify the object. However, though science may have adopted a common human perception pattern to a certain degree it certainly needs a few more steps to make this part of a scientific methodology. In his article Julian Rohrhuber refers to to Bruno Latour who accompanied an expedition for the exploration and research of the soils in northern Brazil as a scientific researcher. He describes simple research methods used for the classification of different kinds of soil based on the colours of different samples of soil[2], soil which in itself describes a complex, constantly changing world (Bruno Latour 1996). Given, a sufficient number of samples exist, these samples allow to draw epistemic conclusions that reflect the situation as a whole.

In our case, when using VideOSC to create control messages for sound or other media, the situation may be somewhat similar. A full resolution image, possibly containing millions of pixels, will certainly overcharge

every processor when trying to create OSC messages from its full colour information. And even if it was possible it would probably be too much information for any kind musical structure to work with. Hence, reduction but yet keeping a significant amount of information is necessary. It is a bit like painting with a big brush: Some details will get lost. Yet it is possible to depict the object in a way that allows the spectator to easily identify what has been painted.

Thus, while the reduction of information during painting is a human decision met in the creative process, it is automated within VideOSC. Human interaction within VideOSC basically means determining the image section. In reference to the French science philosopher Henri Margenau[3] Julian Rohrhuber in his paper makes a distinction between two elements: The epistemic (which is characterized by an operational correspondence with the measurement procedure) and the constitutive, formal component that concerns all other facts[4].

Applied to a system like VideOSC, the connection with some other computational device represents as a whole the data processing unit, the operational correspondence or operational chain, whereas the user interface takes the part of our “samples of soil”, representing visual facts of our perception. A computer or logical unit is necessarily based on a strictly defined operational chain that puts the formal facts into – in our case – musical context. The symmetry within this relationship is based on a simplification that reduces information about reality to a simple pixel pattern. Or in other words, as Rohrhuber puts it in his text, a simple symmetry between a visual field and the common creation process is introduced which bridges the 'abyss' that incorporates the total of all other facts which we do not consider in our musical creation process.

Practical Considerations

VideOSC sends the RGB data of the incoming video with every frame update. Even though the video, respectively the images in each frame are scaled down to a very small size, this still means an enormous amount of data. E. g. at a resolution of 5 x 3 pixels (fig. 2) this still means 45 different values at an update rate of 10-40 frames per second (the rate depends very much on the device's CPU capabilities).

Though the values of OSC messages may seem chaotic or stochastic over time they are by no means random. Every single value is determined by the colour of the pixel as “seen” by the device's camera. However, the special nature of VideOSC's output specially designed sound generating structure.

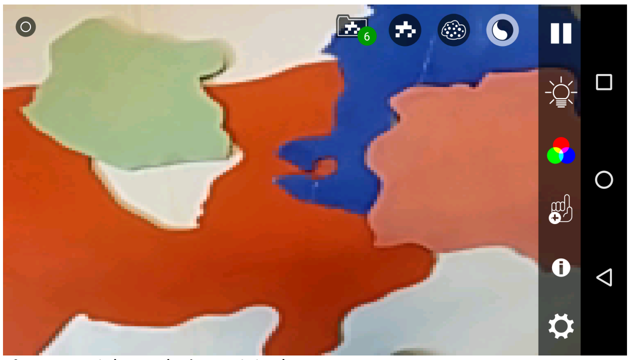

Figure 1. High resolution original

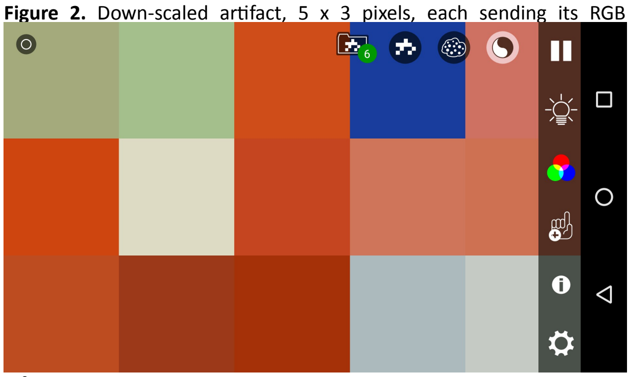

Figure 2. Down-scaled artifact, 5 x 3 pixels, each sending its RGB information via OSC

Most notably, when working with VideOSC, is the fact that all pixels will update with each new frame. Nevertheless, directing the camera at the same image section, should produce the same sound characteristics (given the sound generating synthesis structure does not involve random parameters) but even a slight deviation in the image section will be clearly audible.

Correlation of Colours

The tight correlation within the control determining pixels is yet accompanied by an even stricter correlation between the colour channels of each pixel: A single colour produces three distinct values. A shade of grey for instance will produce the same values for each colour channel in a pixel whereas the colours red, green, blue

will produce high values in their respective channels and low ones in the others. The scheme may be best understood by looking at figure 3. (Of course, except from the basic colours as displayed in figure 3 all other combinations of red, green, blue are possible as well within VideOSC).

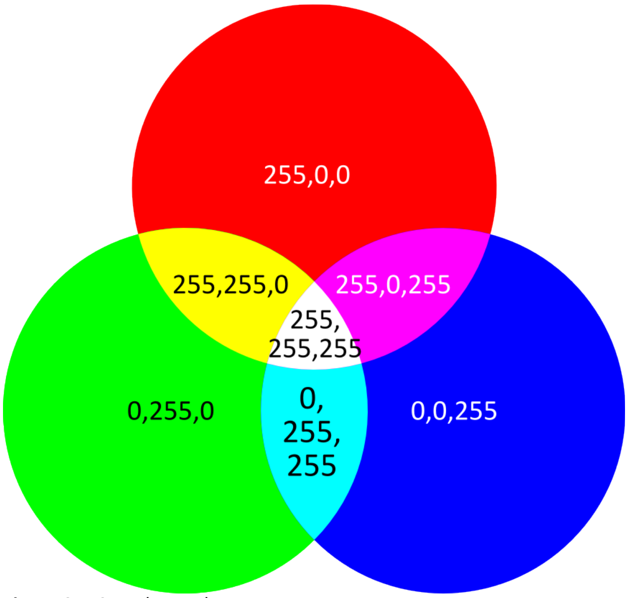

Figure 3. RGB colour values

Hence, the decision to map a control parameter to e. g. the blue channel and another one to the green channel of one and the same pixel may be of special interest.

(Sound) Synthesis Design

VideOSC, beyond creating its complex matrix, does not have any calculating capabilities that would give room for sound design concepts. That has to happen on the listening machine. In general it should be possible to use VideOSC with many different applications. As a minimum requirements those applications must implement the OSC protocol. However, it will require a flexible environment that allows the user to set its parameters in a meaningful relation to values from VideOSC, such as e. g. Pure Data, ChucK and SuperCollider or Max/MSP.

In the following I would to explain a bit my personal approach, using SuperCollider[5].

SuperCollider's audio synthesis essentially involves two essential parts:

1. The design of a synthesis structure (a SynthDef) happens in sclang, the programming language embedded in SuperCollider. A SynthDef is a combination of Ugens (unit generators, e. g. various oscillators or generators that act like unary and binary operators, filters, generators for audio buffer handling, etc.) serves as a blueprint for any
2. Synth instance, the sound producing unit running on SuperCollider's sound synthesis engine, either scsynth or supernova.

The previously described two parts are necessary parts in any sound creation process. The creation of a SynthDef may be done explicitly or it may happen behind the scenes, hidden from the user as well as the instantiation of a new Synth. A Synth will live in the sound synthesis engine, either scsynth or supernova. Nevertheless, a Synth, once it is playing, is absolutely sclang agnostic and can only be addressed via OSC commands.

Running Synths on a server are organized in a tree-like structure. Each branching (a node) is uniquely identified by an integer id. To effectively address a running Synth it must be addressed by the enclosing node's id. Also the tree structure determines which output plays to which input, e. g. when instantiating a Synth that acts as a filter for the output of another Synth. The class Synth implements a few useful commands (wrapping pure OSC commands in a more user friendly syntax) to make handling of running Synths easier, yet it does not allow any reorganization of nodes once a Synth has been instantiated on the server.

The previously written makes it evident that using generic Synths only is not really convenient, especially in situations when a user wants to quickly reorganize an already playing synthesis struture on the server. Therefore SuperCollider, respectively sclang, implements a number of high-level structures that may handle these tasks in a more convenient way. E. g.:

- a NodeProxy, a ProxySpace or an Ndef – all of these are basically the same thing with a different flavor and act as containers for sound synthesis structures in a similar way as SynthDefs. Beyond that they do also handle the creation and ordering of nodes on the server. Also they can be rewritten on the fly and an already existing node structure on the server will get updated repsectively replaced accordingly. NodeProxy, ProxySpace and Ndef instances may be nested. They may embed other structures like
- a Pdef, another proxy structure, acting as a container for Patterns, a special group of sclang classes that allow directives for timed execution of Synths, either setting control inputs of already running Synths or instantiating new ones (granular synthesis) in defined sequences. The Pdef will pass on new node ids to it's enclosing NodeProxy / ProxySpace or Ndef which in return will take care of the correct node ordering respectively structure. Pdefs can be rewritten on the fly as well.

Using the previously described concepts, complex synthesis structures including sequencing can be defined quickly in a flexible manner. Yet, what is missing is a layer that gives the user full control over all elements and provides connectivity with external hardware, such as

MIDI and OSC capable devices. Therefore I have written my own sclang library named CVCenter[6] (see a screenshot in figure 4). CVCenter is basically a collection of CVs[7] (instances of CV), a low level object that holds a numeric value or an array of numeric values constrained by a ControlSpec which itself defines a ramp between two values and a curve parameter, defining how values will grow from low to high (usually linear or exponential). Additionally an arbitrary number of Functions (directives that can be executed on demand) can be added as dependants to a CV at any time. The Functions will get executed every time the value of the CV is updated. CVCenter itself enhances the functionality of a CV with a graphical user interface and the ability to quickly connect external MIDI or OSC capable devices, either through code or the graphical user interface. A setup of CVs respectively CVWidgets in CVCenter can be stored to disc including Functions and current MIDI/OSC connections and can be restored at any time.

Yet, probably most important, CVCenter has the ability to analyse the structure of any running Synth, Ndef, NodeProxy or ProxySpace and can automatically create a graphical user interface, allowing to set all controls via mouse or external MIDI/OSC applications and devices. Pdef in turn allows the embedding of CVs directly in its notation. A numeric value or another Pattern can directly be replaced by a CV.

Figure 4. CVCenter, automatically derived from a running Synth

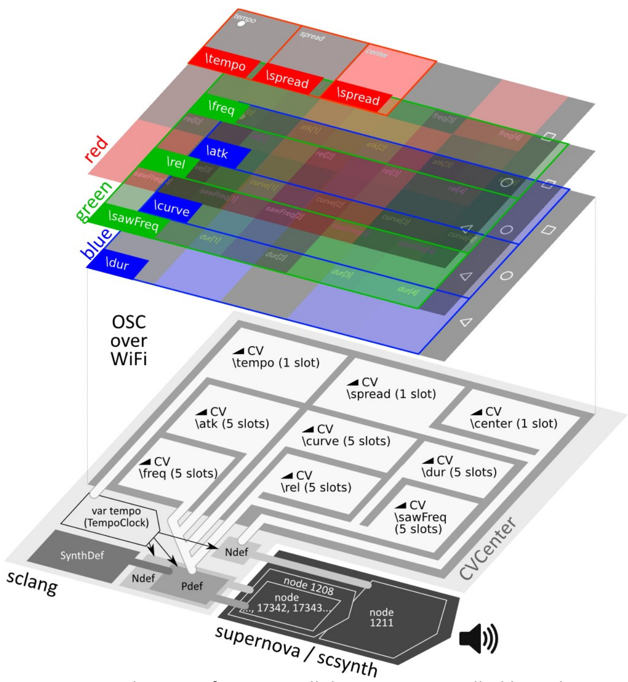

Figure 5. Visualization of a SuperCollider setup, controlled by VideOSC - the upper three layers (red, green, blue) shows the distribution of controls accross the pixels (5 x 3). The lower layer displays the various elements within the SuperCollider setup. The illustration simplifies the configuration a bit. Yet it should clearly demonstrate the tight relationship between CVs and the VideOSC interface as well as the various relationships between CVs and other parts running in SuperCollider

Conclusion

Due to its technical concept VideOSC allows fine grained control over complex sound structures. Despite its deterministic output (the values being sent to the receiving client as OSC messages) usage and handling of VideOSC will differ much to other control applications and devices. Where normally manual interaction or gesture controls the sound parameters it is the composite information of a complex matrix of values that controls all parameters at the same time.

VideOSC is an attempt to explore the visual appearance of our world for its specific qualities and possibilities. Where a painter translates a visual impression into a painting it translates the visual into sound (or some other expression of electronic media). The analogy of the “electronic brush” seems to be nearby. Yet, where the painter attempts to create an image of the world as she or he sees it, VideOSC puts a strong constraint on the protagonist as the application already yields a fully defined image. Nevertheless it is only a small detail and leaves as many options as there are possible angles of

view on this world. Or even beyond that: It invites the performer to actively interfere with appearance of what she or he sees, to interact with and modify the image not inside the application but outside in the real physical world.

Figure 6. Kazimir Malevich, Black Square (one of several versions). Foto taken 2007 at Hermitage Museum, St. Petersburg

As a result VideOSC asks the questions about similarities in artistic visual and acoustic processes. Within the long mimetic tradition in the European history of art it was possibly Kazimir Malevich whose "Black Square" marked a final culmination – an image revealing an essential quality of reality: information. Likely Malevich did not have digital respectively the binary aspect in mind when he created the first version of his famous painting. Yet he was certainly aware of of the iconic character his work had in respect of the mimetic nature of art. The art theorist Philip Shaw writes the following:

> "What Malevich's painting does is 'simply render – or isolate – this place as such, an empty place (or frame) with the proto-magic property of transforming any object that finds itself in its scope', even a black square of pigment, 'into a work of art'" (Philip Shaw, January 2013).

In analogy to the previous quote one might say it is the "proto-magic" property of the pixel that may transform anything within its scope to sound or some other form of electronic media.

References

Julian Rohrhuber, "Operator, Operation – Seeing what the Photon sees", in Strukturentstehung durch Verflechtung. Akteur-Netzwerk-Theorie(n) und Automatismen, Fink Verlag, Munich 2011, retrieved from http://www.wertlos.org/~rohrhuber/articles/Rohrhuber_Operation_Operator_Sehen_was_das_Photon_sieht.pdf in July 2016

Bruno Latour, "Der 'Pedalogen-Faden' von Boa Vista – eine photo-philosophische Montage"; in Der Berliner Schlüssel, Erkundungen eines Liebhabers der Wissenschaften, Berlin 1996, p. 191-248

Henry Margenau, The Nature of Physical Reality, 1950, p. 237

Philip Shaw, "Kasimir Malevich's Black Square", in Nigel Llewellyn and Christine Riding (eds.), The Art of the Sublime, Tate Research Publication, January 2013, retrieved from https://www.tate.org.uk/art/research-publications/the-sublime/philip-shaw-kasimir-malevichs-black-square-r1141459, accessed 22 July 2016.

1 Open Sound Control – a protocol for communication among computers, sound synthesizers, and other multimedia devices. http://opensoundcontrol.org

2 Bruno Latour, “Der 'Pedalogen-Faden' von Boa Vista – eine photo-philosophische Montage”; in Der Berliner Schlüssel, Erkundungen eines Liebhabers der Wissenschaften, Berlin 1996, p. 191-248

3 Henry Margenau, The Nature of Physical Reality, 1950, p. 237

4 “Um sowohl diese logische als auch die operationale Beziehung in sein System zu integrieren, spricht Margenau von zwei Formen der Definition, nämlich der epistemischen (die eine operationalisierbare Korrespondenz zu einem Messverfahren hat) und der konstitutiven, bzw. formalen (die alle anderen Tatsachen betrifft).”, in Strukturentstehung durch Verflechtung. Akteur-Netzwerk-Theorie(n) und Automatismen, Fink Verlag, Munich 2011, p. 85

5 As an object orientated language SuperCollider's inbuilt programming language sclang is organized in classes. Classes are named with words beginning with a capital letter. The class system is hierarchically organized. All classes inherit from a base class named Object. sclang, scsynth and supernova, though they denote names, are written in small letters. The word server may refer to scsynth as well as supernova.

6 https://github.com/nuss/CVCenter

7 CV is part of the Conductor library, written by Ronald Kuivila. The current version of CVCenter uses a slightly modified version of the library that can be found at https://github.com/nuss/Conductor